DOCUMENT RESUME

ED 262 757 IR 011 835

TITLE	Article Reprints from "The Computing Teacher" on Software Evaluations.
INSTITUTION	International Council for Computers in Education, Eugene, Oreg.
PUB DATE	85
NOTE	25p.
AVAILABLE FROM	International Council for Computers in Education, University of Oregon, 1787 Agate Street, Eugene, OR 97403.
PUB TYPE	Collected Works - General (020) -- Guides - Non-Classroom Use (055) -- Viewpoints (120)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Computer Assisted Instruction; *Computer Software; Curriculum Development; Evaluation Methods; *Instructional Material Evaluation; Instructional Materials; *Media Selection; Microcomputers; Science Education
IDENTIFIERS	Computer Uses in Education; *Software Evaluation

ABSTRACT

Reprinted from the "The Computing Teacher," this collection of nine articles presents information on computer software selection and evaluation. The articles include: (1) "The DISC Model for Software Evaluation and Support Material Design" (Shelley Yorke Rose and Carol Klenow); (2) "Selecting Computer Software--We Take It Seriously" (Jean Donham); (3) "How Do Teacher and Student Evaluations of CAI Software Compare?" (Barbara Signer); (4) "The Software Selection Process: Some Management Questions" (Anne Batey); (5) "Enrichment Courseware for Middle School Mathematics" (Lois Edwards); (6) "Computers in Science Education" (Richard C. Adams); (7) "Doing Science" (Richard C. Adams); (8) "Statistically Speaking" (Carl Edeburn); and (9) "Bank Street Writer--Does the Tool Define the Process or Vice Versa?" (Randall Boone). Individual articles include a list of approved instructional software for the Iowa City Community School District (Donham); a sample student software questionnaire (Signer); and reviews of sample science education software (Adams). (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED262757

ARTICLE REPRINTS

From the "Computing Teacher"

on

Software Evaluations

BEST COPY AVAILABLE

IR011835

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

David Morsund

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The DISC Model for Software Evaluation and Support Material Design

by
Shelley Yorke Rose and Carol Klenow

Educators in Michigan now realize that they should preview and evaluate software prior to purchase.

Evaluation allows teachers to choose bug-free software which is based on accepted instructional principles and written for specific subject areas and grade levels. Previewing allows teachers to plan activities that complement the software and to devise ways to measure student learning.

While evaluation prior to purchase is a major step toward acceptance and implementation of computerized instruction in our schools, the process needs to be supplemented to enhance classroom use.

"The support material components were based on the DISC Project's philosophy that successful use of software required teacher planning, background information, pre- and post-computer activities and a record-keeping device."

The DISC (Documentation and Integration of Software into the Classroom) Project has increased the effective use of educational software in Oakland County by creating a model for instructional support materials to accompany evaluated educational software while training a network of 100 educators in its application.

This was accomplished by a two-phase approach. First, DISC participants were trained in the use of an adapted MicroSIFT evaluation form. They applied MicroSIFT's criteria to an educational software package purchased for them by the DISC Project. Educators then underwent the second phase of training in use of the DISC model. The software which participants recommended for classroom use was targeted for design of accompanying support materials which would facilitate teacher planning and increase its educational value to students.

The support material components were based on the DISC Project's philosophy that successful use of software required teacher planning, background information, pre- and post-computer activities and a record-keeping device. These requirements for effective use led to the design of the six components of the DISC support materials package.

Before using software, an educator should first assess whether available classroom time, setting and student abilities are compatible with a program's delivery of instruction. Next, the educator requires information regarding the program's educational content. In DISC's *Guidelines for Teachers*, this information is referenced in a one-page overview which precedes the program operation observations recorded in the *Program Operation Procedures* section.

Next, *Teacher Lesson Plans* encourage DISC participants to consider how the program would fit into a unit of instruction. Finally, the three remaining components are directed toward student use of the program. The *Preliminary Worksheet for Students* was designed to relate the computer activity to classroom objectives. This activity seeks to contravene the popular notion that computer activities do not reinforce traditional curriculum objectives. The *Reinforcement Worksheet for Students* is similarly designed to follow up computer use with activities such as class discussion or research projects. Finally, a *Computer Activity Worksheet* enables the teacher to monitor student interaction with the computer.

Participant feedback, the high quality of the support materials contributed to the *DISC Compendium*, and the continued interest in the DISC model have indicated the model's viability and acceptance both locally and nationally.

A similar project need not be expensive to implement. The majority of DISC's $15,000 funding went toward payment of substitute teachers while participants attended four 1/2 day sessions. The modest investment for software per participant (under $30) was generously repaid with support materials invaluable to many other software users.

The institution of the DISC model also saves money by preventing duplication of efforts. For us this meant that users of the more than 2,000 microcomputers in our 28 school districts could consult the *DISC Compendium* prior to software purchase and classroom implementation. These teachers then were provided not only with an evaluation, but also

© Copyright 1985 ICCE

with ready-made materials for adoption or adaptation *and* with a contact person experienced in the use of that software package. In an environment of experienced educators who have become wary of educational fads, support materials that relate software to curriculum objectives positively affect the establishment of the computer as an instructional tool.

"In an environment of experienced educators who have become wary of educational fads, support materials that relate software to curriculum objectives positively affect the establishment of the computer as an instructional tool."

A collection of evaluations and support materials (in this case, the *DISC Compendium*) simplifies individual curricular decisions while streamlining instructional use of microcomputers. This is especially significant in districts with many novice users to whom the computer's classroom role is not yet clearly defined. Just as new textbook users rely heavily at first on the instructor's manual, so do novice computer-using educators rely on software documentation. At times, experienced computer-using educators become novices when facing a new software package or new hardware.

A project adopting the DISC model should instruct educators in the importance of reviewing both evaluations by peers and reviewing commercial or teacher-developed support materials prior to judging the viability of a software package in their classrooms.

With DISC-based training to improve their software selection and implementation skills, educators will find that the microcomputer can be used at last to make a difference in the educational process.

[Ed. note: Carol Klenow and Shelley Yorke Rose are Instructional Computing Consultants for the Interactive and Instructional Computing Department (IICD) of the Data Processing Division of Oakland Schools, an intermediate school district serving 28 local school districts with over 200,000 students in Oakland County, Michigan.

For additional information on the DISC Project, contact the IICD-Oakland Schools, 2100 Pontiac Lake Road, Pontiac, MI 48054. Copies of the DISC Compendium *may be purchased through the IICD for $20. Also available is the* DISC Project Report and Training Manual *($10) which includes session outlines, hand-outs and transparency masters.]*

END■

© Copyright 1985 ICCE

Selecting Computer Software—We Take It Seriously!

by
Jean Donham

The Iowa City Community School District serves 8300 students in 20 schools. The district developed a plan for instructional uses of microcomputers in 1982. The plan is being implemented in phases, based on available dollars and opportunities for teacher inservice.

One component of this initial plan allocated $5000 per year for the first three years to purchase software for evaluation by the selection committee and by the rest of the teaching staff. The district selected the *Evaluator's Guide for Microcomputer-based Instructional Packages*[1] as the evaluation instrument because it gives a comprehensive set of criteria.

The Process

The Superintendent directed the district library media coordinator to appoint a district-level committee to select software. This 20-member committee includes teachers, library media specialists and administrators. They were chosen because of their familiarity with computer-assisted learning and their knowledge of instructional design; the committee encompasses a range of content-area specialties. The district library media coordinator chairs the committee.

Teachers and library media specialists throughout the district suggest software for the committee to consider. If a title has already been previewed, a copy of the completed MicroSIFT evaluation form is sent to the person making the request. If a title has not been evaluated, the software is ordered on approval. (If a company refuses to sell on approval, their product is not considered for purchase.) Software is usually kept in the district on preview for one month.

When the software arrives, three committee members are assigned to evaluate the software. Evaluations are done individually in each evaluator's school so that students can be involved in trying out the software when appropriate. While effective software design is an important consideration, the evaluators also compare the objectives of the software to the district curriculum. This is an essential consideration and sometimes results in very well-designed software being rejected. (This has been a frequent problem, particularly in language arts and social studies.) Each evaluator uses the MicroSIFT criteria to assess the software and makes a recommendation either to purchase or not to purchase. The three evaluations are then reviewed by the coordinator, who in many instances also evaluates the software before making a final decision.

If approved, one copy of the software is purchased out of the central fund and is then available for preview by all schools through the District Media Center. All schools receive quarterly annotated lists of approved software. Schools may only purchase approved software (see Appendix).

If a software title is rejected by the committee, a copy of the completed evaluation form is sent back to the company along with the returned software. This practice provides producers with feedback. To date, over 275 instructional software titles have been previewed. Nearly two-thirds of the software evaluated has been returned.

Two *Core Software Library* lists, one for elementary and one for junior high (see figures 1 and 2) have also been developed by the software selection com-

Core Software Collection—Elementary Level
May 1984

Bank Street Writer
Scholastic

Basic Number Facts
Control Data

Delta Drawing
Spinnaker

Gertrude's Puzzles
Learning Company

Jigsaw
Island Software

King's Rule
Sunburst

Master Type
Lightning Software

Memory: A First Step
Sunburst

Teasers by Tobbs
Sunburst

Figure 1.

Core Software Collection—Junior High Level
May 1984

Applewriter for II + or IIe
Apple Computer, Inc.

Archaeology Search
McGraw-Hill

Bank Street Writer
Scholastic

The Factory
Sunburst

French Vocabulary Builder
Control Data

Lincoln's Decisions
Educational Activities

Master Type
Lightning Software

Nutrition and Nutrition Simulation
Health Education Services

Rocky's Boots
Learning Company

Spanish Vocabulary Builder
Control Data

Terrapin Logo
Terrapin, Inc.

Tobbs Learns Algebra
Sunburst

Typing Tutor
Microsoft

Word Attack
Davidson & Associates

Figure 2.

[1] *Evaluator's Guide for Microcomputer-based Instructional Packages*, MicroSIFT, Northwest Regional Educational Laboratory. Published by ICCE.

© Copyright 1985 ICCE

mittee. These titles were identified as appropriate for all elementary and junior high schools to own and use, based both on design and relationship to the curriculum. The list includes word processing and CAL software for content areas, and a Logo package. These libraries have been purchased for all schools in the district at approximately $600 per school.

Discussion

This approach to software selection has many strengths. First, it allows for independent evaluations. Evaluators work in their own schools rather than coming together to review software. This reduces the likelihood that one person's point of view will influence the decision making of another.

In addition, since evaluators work in their own schools to review software, they are able to try the software with students. Student interaction with a program often helps identify its strengths or weaknesses. This is especially true when considering ease of use, clarity of instructions, level of difficulty and readability of the program.

Software evaluation is a time-consuming task. Having a process coordinated at the district level has some benefits:

- The total amount of teacher time that might be spent in evaluating software is reduced. Teachers *not* on the selection committee spend their time reviewing software already identified as well-designed and appropriate. They can then move on to the task of determining how to use that software effectively with their students.
- A central file of evaluation results prevents repeated evaluations of the same software.

Saving teacher time carries with it some cost, however. Approximately 20% of the library media coordinator's time is spent on the software evaluation process. In addition, clerical time at the central office is required for ordering, returning and record keeping. Still, the process is efficient.

Responsible software evaluation is critical if teachers are expected to subscribe to the value of computer-assisted learning and if students are to benefit from it. Further, producers need to know that educators have high standards for acceptable software. Indeed, in Iowa City, we take software evaluation very seriously. **END**■

Appendix
Approved Instructional Software
Iowa City Community School District
May 1984

Software approved for use in this district has been reviewed by the selection committee on the basis of its sound design and its appropriateness for Iowa City's curriculum. Please note: The titles listed below do not include the titles in core libraries. The indicated grade levels are the grades at which the software is used in Iowa City.

Title	Producer	Subject	Level
Africa	Educational Activities	Social Studies	Grades 7 up
Algebra Drill & Practice I	Conduit	Mathematics	Senior high
Asean	Educational Activities	Social Studies	Grades 7-8
Atomic Structure	Opportunities for Learning	Science	Senior high
Basic Math Facts Drill	Random	Mathematics	Grades 4-8
Bumble Games	Learning Company	Mathematics	Grades 2 up
Bumble Plot	Learning Company	Mathematics	Grades 3 up
Cellab	Conduit	Biology	Senior high
Challenge Math	Sunburst	Mathematics	Grades 7 up
Chem Lab Simulation #1	High Technology	Chemistry	Senior high
Clock	Hartley	Telling Time	Grades 2-3
Color Keys	Sunburst	Problem Solving	Grades 4 up
Computer Math Games	Addison Wesley	Mathematics	Grades 1 up
Dasher	Conduit	Foreign Language	Senior high
Decimals	Control Data	Mathematics	Grades 7-8
Demographics	Conduit	Social Studies	Senior high
Demolition Division	DLM	Mathematics	Grades 4 up
Dragon Mix	DLM	Mathematics	Grades 4 up
Eat Smart	Pillsbury	Home Economics	Grades 7 up
Experiments in Human Physiology	Human Relations Media	Science	Senior high
Explorer Metrix	Sunburst	Mathematics	Grades 7-8
Exploring Logo	Sunburst	Problem Solving	Grades 4 up
Factoring	MicroPower & Light	Mathematics	Grades 7-12
French Travel Vocabulary	Control Data	French	Grades 9 up
French Vocabulary for Shopping Use	Control Data	French	Grades 9 up
Fundamental Skills for General Chemistry	Opportunities for Learning	Chemistry	Senior high
Gas Laws & Kinetic Molecular Theory	Merlan Scientific	Science	Senior high
Geology Search	McGraw-Hill	Science	Grades 7-8
German Classroom Words	Control Data	German	Grades 9 up
German Vocabulary Building	Control Data	German	Senior high
Green Globs	Conduit	Mathematics	Grades 7 up
Incredible Laboratory	Sunburst	Problem Solving	Grades 3 up
Juggler's Rainbows	Learning Company	Spatial Concepts	Grades K-2
Kid Stuff I & II	Circadian	Computer Programming	Senior high
Le Francais Par Ordinateur —Paris en Metro	Heath	French	Senior high
Le Francais Par Ordinateur —Un Repas Francais	Heath	French	Senior high
Managing Lifestyles	Sunburst	Mathematics	Grades 7 up
Mastering the Parts of Speech	SVE	Language Arts	Grades 7 up
Math Ideas with Base 10 Blocks	Cuisenaire	Mathematics	Grades 2 up
Mathematics Activities Courseware	Houghton Mifflin	Mathematics	Grades 4-6
Meter Multiplication	DLM	Mathematics	Grades 4 up
Physics—Elementary Mechanics	Control Data	Physics	Senior high
Problem Solving in Algebra	Encyclopedia Britannica	Mathematics	Senior high
Reading Comprehension I	Milton Bradley	Reading	Grades 5-8
Robot War	Muse	Computer Programming	Senior high
Sentence Combining	Milliken	Language Arts	Grades 4-8
Spanish Classroom Words	Control Data	Spanish	Grades 9 up
Spanish Travel Vocabulary	Control Data	Spanish	Grades 9 up
Spanish Vocabulary for Shopping Use	Control Data	Spanish	Grades 9 up
Speed Reader	Davidson & Assoc.	Language Arts	Grades 9 up
Step by Step 2	SVE	Computer Programming	Senior high
Survival Math	Sunburst	Mathematics	Grades 7 up
Tutorial Quiz Master	Random	Interdisciplinary	Grades 3 up
Typing Tutor II	MicroSoft	Business	Grades 8 up
Waves and Vibrations	Merlan Scientific	Physics	Senior high
Word Master	DLM	Language Arts	Grades 3 up

[Jean Donham, Library Media Coordinator, Iowa City Community School District, Iowa City, Iowa.]

© Copyright 1985 ICCE

BEST COPY AVAILABLE

How Do Teacher and Student Evaluations of CAI Software Compare?

by
Barbara Signer

INTRODUCTION

Since the 1960s, the literature contains much research on computer assisted instruction. Traditionally, the studies have examined the following:

1. Effects of CAI on academic achievement.
2. Effects of CAI on attitudes toward subject matter.
3. Effects of CAI on attitudes toward computers.

More recently, studies have explored:

1. Needs for computer awareness.
2. Needs of the teacher in the CAI classroom.
3. Description and formative evaluations of computer projects in education.

Studies that compare student and teacher perceptions of CAI software are lacking. As an evaluator for a State of Florida grant for CAI research, I included this issue in the evaluation study.

DESCRIPTION OF CAI PROJECT EVALUATED

The grant developers proposed a unique alternative for the microcomputer as an instructional tool. It integrated the microcomputer into the conventional classroom in conjunction with the teacher's own teaching style—that is, the teacher controlled the pace and parameters of the program. One machine was required per classroom, and viewing was on a 19-inch screen. Interaction and opportunities for discussion were intended for total class presentation. The microcomputer was not used for every lesson, but only where it complemented rather than duplicated instruction by the teacher or texts. Therefore, instead of traditional usage of the term CAI, the project used computer integrated instruction (CII). The student data source consisted of students enrolled in all five Algebra II classes at a high school where CII was implemented. The teacher data source consisted of the two Algebra II teachers.

INSTRUMENTS

To obtain information necessary to evaluate the perceived effectiveness of CII, a variety of instruments were needed. If possible, an existing CAI evaluation instrument was used or modified; otherwise, one had to be developed specifically for this evaluation. The instruments used were the following: a) Signer Student Software Checklist, b) Signer Teacher Software Questionnaire and c) NCTM Teacher Software Evaluation Checklist.

Student Software Evaluation Checklist

The function of the *Signer Student Software Evaluation Checklist* (Figure 1) was to provide the evaluator with student feedback on the motivational and instructional value of the software used. This feedback included both teacher directed and independent student use. The criteria included for the instrument were: a) ease of administration, b) codability of items, c) opportunity for comments for each item and d) ability to identify strengths and weaknesses. While many questionnaires exist to solicit these kinds of information from students, all are intended for individual student use of computers only, and are not applicable for CII. In light of this, an original checklist was designed for this study. A content validity check of the instrument was accomplished by having specialists of mathematics education and educational evaluation review the items in the checklist for completeness of criteria and utility of information gathered through the instrument. The schedule of use of the checklist varied with the use of the software. Programs could be used one time or several times, by the class or by an individual. When instruction was presented to the class as a whole, two randomly chosen students from each class completed the checklist for each lesson. When programs were used by students individually, a checklist was completed by each student. Programs were evaluated at the time of use throughout the study.

Teacher Software Questionnaire

The aim of the *Signer Teacher Software Questionnaire* (Figure 2) was to collect teacher input regarding the value of the software for classroom presentation. The focus of the checklist included the following: a) classroom management, b) amount of teacher direction, c) amount of teacher preparation, d) suggested uses and e) a comparison to traditional instruction. The questionnaire included items that were open-ended so that software strengths and weaknesses could be identified.

Since this instrument evaluated a unique application of computer assisted instruction, the evaluator could not use available validated instruments to gather the information. A content validity check on

© Copyright 1985 ICCE

the newly developed instrument was conducted by having the two specialists, one in mathematics education and one in educational evaluation, review the overall questionnaire and determine whether items in the questionnaire solicited relevant feedback from teachers. Items were revised based upon reviewers' comments. It was used in conjunction with the Teacher Software Evaluation Checklist. Both forms were completed by each teacher after the initial use of the program.

NCTM Teacher Software Evaluation Checklist

The purpose of the NCTM Teacher Software Evaluation Checklist was to gather teacher feedback on the CII software. Criteria used to evaluate the programs were the following: a) appropriateness of level of instruction, b) diversity of modes, c) program flow and logic, d) motivational value and e) use of computer capabilities.

The *NCTM Software Evaluation Checklist* is contained in the National Council of Teachers of Mathematics (NCTM) publication, *Guidelines for Evaluating Computerized Instructional Materials*, prepared by Heck, Johnson and Kansky (1981). It contains items consistent with the criteria of the study and could be applied directly to the CII environment. The designing and field testing of the checklist were performed by a special NCTM Instructional Affairs Committee (IAC), chaired by Robert J. Kansky of the University of Wyoming. During this study, the checklist was completed by each teacher after initial classroom presentation of the software.

RESULTS

Teacher ratings were tabulated from the Software Checklist and the Teacher Questionnaire. Student opinion of each program reflected a consensus of the Student Questionnaires. The modal response for each question was recorded per program when clearly predominant. When the responses varied in both the positive and negative direction, the value assigned was recorded as mixed. Comments were noted when they occurred frequently on the evaluation instruments. Teacher ratings were slightly above student ratings for both instructional and motivational value. Teachers rated these criteria high across the programs. Students rated a small number either moderate or low on these criteria. There was a consistent difference between teachers and students when rating a program's pace. The teachers' ratings were favorable, while the students' were poor. Teacher-recommended changes were specific and unique to the particular program, while student-recommended changes were generalizable across programs, stressing more individual student use. Both teachers and students focused on motivational and visual qualities (graphics) for program strengths.

The reported differences between teacher and student evaluations can be attributed to their different roles. Teachers, as content specialists, are more critical of the specific content of a program. Knowledgeable in teaching strategies and the development of the concepts, they are aware of the varying pedagogical complexities for the different topics. Students, not cognizant of these intricacies, evaluate programs on the basis of interest, clarity and their level of participation.

LIMITATIONS

Conclusions of this evaluation study are not generalizable to all computer assisted instruction (CAI) studies. Computer integrated instruction (CII) is a unique application of computer assisted instruction. Conclusions are also specific to the environmental constraints of the school, which include fiscal resources and teacher personalities. These factors contribute to the effectiveness of teacher training. As teachers receive more and better training their perceptions of software will become more critical.

CONCLUSIONS

The natural implication of this study is that students and teachers have different perceptions of quality software, with the students being the stronger critics. Today's students are not naive about computer capabilities in terms of graphics and sophisticated peripherals. Their environment is filled with three-dimensional computer games with sound. Just as schools have had to compete with television, educational software must compete with the software found in arcades. As adults, we often assume that a microcomputer is motivating by itself. This is evident with the abundance of drill and practice and page-turning programs. Interactive programs, based on sound pedagogical development, require the feedback of both students and teachers. Extensive field testing in the schools is needed to move closer to higher quality software.

FUTURE RESEARCH

Educational researchers agree that formative evaluation is critical for successful development of innovations. This study attempted to answer the question: How do student and teacher evaluations of computer materials compare? The author suggests this and the following questions be included in future studies:

1. How do teacher and observer evaluations of computer use compare?
2. What are the teacher identified needs and levels of support to improve computer use?
3. What are the student perceived problems, their causes and recommendations?

The author will share details of this study. Those interested in future cooperative research that ad-

© Copyright 1983 ICCE

dress the above questions are encouraged to write the author.

REFERENCES

National Council of Teachers of Mathematics. *Guidelines for Evaluating Computerized Instructional Materials.* Reston, VA: National Council of Teachers of Mathematics, 1981.

Signer, B.R. *A Formative and Summative Evaluation Study of a Project Integrating the Microcomputer with Second Year High School Algebra Instructors,* doctoral dissertation, University of South Florida, 1981.

Figure 1.
Signer Student Software Questionnaire
[Write author for additional validation data and permission to reproduce forms.]

This will not be graded. Your comment will simply be used to improve your instruction. Your help in evaluating the instruction is invaluable.

Date _____ Evaluator's Name _____

Program Name _____

Chapter _____ Section _____

T_____ F_____ 1. I enjoyed the microcomputer program.
(Comments)

T_____ F_____ 2. Working with the screen became boring.
(Comments)

T_____ F_____ 3. I found myself just watching the screen, rather than trying to learn.
(Comments)

T_____ F_____ 4. I feel satisfied with what I learned.
(Comments)

5. The pace of the instruction using the microcomputer was:
_____ Just right _____ Too fast _____ Too slow
(Comments)

6. The amount of material covered during the lesson was:
_____ Just right _____ Too much _____ Not enough
(Comments)

7. I feel that the material presented was:
_____ Too difficult for me
_____ Just normal for me
_____ Perfect, I really learned quicker with the microcomputer
_____ Mostly a review since I already knew it
(Comments)

8. What would you change to make this lesson better for you?

9. What did you really like about this lesson?

Figure 2.
Signer Teacher Software Evaluation Checklist

Program _____ Microcomputer _____
Lesson Topic _____ Ch _____ Sec. _____
Teacher _____ Program Author _____
Times Presented _____

1. Was the pace appropriate for your instruction? (Why or why not?)

2. What would you like to see altered?

3. What suggestions do you have for teachers using it for the first time?

4. Was outside preparation required?

a. How much time?

b. Preparation of additional materials?

5. Compare this lesson using the microcomputer program to your past teaching of the topic.

a. Student behavior

b. Facilitation of material to be learned

6. General comments (hardware, classroom management, etc.).

7. Program use: Introductory? Drill practice? Review?

a. Your use

b. Suggested use

END■

Barbara Signer, University of Houston, Central Campus, College of Education, Department of Curriculum & Instruction, Houston, TX 77004.

© Copyright 1983 ICCE

The Software Selection Process: Some Management Questions

by
ANNE BATEY

A survey of the literature on software selection reveals an abundance of carefully outlined criteria for thorough software evaluation. Although software evaluation is central, it does not complete the software selection process. Information on how to efficiently organize and manage the selection process is scarce. An exchange of such information would benefit any district that is developing a software selection process. This article poses three questions about management of the software selection process:

- Who is responsible for each step and decision?
- How is information transferred among the personnel involved?
- How long does the process take?

Following a discussion of the questions, an exercise is suggested to help you focus on the management aspects of your district's selection process. The purpose of the exercise is to generate a response from those of you in districts that have working answers to share.

Question 1: Who is (or should be) responsible for each step and decision in the software selection process?

A software selection process requires numerous steps and considerations. No single individual can be responsible for all of the decisions that have to be made before a package is purchased. The process is time consuming and should involve many district personnel who have a range of computer expertise and leadership roles. The personnel involved includes computer coordinators, computer facilitators, classroom teachers and students. The term "computer facilitator" refers to a variety of possible titles and roles—members of a software selection committee, in-building computer leaders, media specialists, teachers of educators, etc. In general, computer facilitators are the computers-in-education specialists. They often have considerable contact with classroom teachers, either because they continue to teach or because their jobs keep them within individual school buildings.

The question of respective roles in the process is refined by considering who is best suited for each responsibility. In general:

1. Computer Coordinators provide direction concerning decisions in the selection process that are administrative in nature, e.g., the budget, long-term planning, or district-wide computer curriculum.
2. The computer facilitators have technical expertise and can address the decisions and responsibilities that involve the implementation of educational software. Their contributions might include the following: educating teachers; researching software catalogs, reviews and evaluations; providing knowledge of the hardware capabilities; and screening for general software performance.
3. Teachers are in the best position to determine if a package meets their classroom needs and objectives. Final decisions that involve the instructional content and effectiveness should belong to the teachers who will be using the package.
4. Student response to a package also provides information for the decisions involving instructional effectiveness.

Question 2: How is information transferred among personnel involved in the software selection process?

Regardless of the size and the computer sophistication of a district, the selection process will generate lots of information. Since the process is not the sole responsibility of one person or group, considerable information flow and feedback must take place. How and by what routes does the information flow?

Question 3: How long does the software selection process take?

Many mistakes have been made in the haste to purchase software for newly-acquired hardware. A careful selection process reduces the waste of purchasing useless and ineffective software. But what is a reasonable time frame? How long do the individual stages in the process take?

Formulating Your Answers

The remainder of the article is an outline of the selection process. It summarizes some essential steps suggested by articles in computer education publications. The list is neither detailed nor comprehensive. It is included to encourage consideration of the management aspects of your own district's software selection process. In the interest of developing and sharing answers to the above questions, the following exercise is suggested:

© Copyright 1985 ICCE

Read through the steps and note the names or titles of the individuals that are responsible for each of the steps. Add, delete or change the sequence of steps as necessary. Some of the steps may involve a combination of personnel. Try drawing lines and arrows to clarify the flow of information through the process. Estimate the amount of time that each step requires. Perhaps the result of the exercise could be organized into a table or a flowchart. You are encouraged to submit your results. The information is needed and worth sharing.

I. **Preliminaries in the selection process:**
- Educate teachers: a software inservice, evening classes, etc.
- Collect resources: the software directories, catalogs, reviews.
- Develop schedule for the selection process.
- Determine budget limitations.
- Become well-versed in the district copyright policy or if necessary, establish one.

II. **Needs and objectives of your curriculum:**
- Develop a list of instructional needs to be met with computer use.
- Clarify needs, e.g., determine number of students involved per use, estimate desired length of involvement, specify the amount of teacher management necessary.
- Develop instructional objectives.

III. **Find software to match curriculum objectives:**
- Shop the catalogs with objectives, target audience, budget and hardware in mind.
- Read descriptions, reviews and evaluations.
- Decide on an acceptable package.
- Obtain preview copy: order from publishers who grant preview periods or provide demo disks.
- Ensure that preview rules and deadlines are followed.

IV. **Initial Software Preview:** A technical screening for performance in areas that do not involve instructional content. This includes:
- Complete user support materials
- Trouble-free operation, well-designed screen layout, sufficient user control
- Descriptive error messages with branching options
- Appropriate use of computer capabilities.

V. **Preview of software content:** A screening for instructional content. This includes:
- Topic and objective are clear in program and documentation.
- Program objective(s) match desired objective(s).
- Computer availability will not limit effective use of package.
- Management demands on teacher will not limit effective use of package.
- Students learn from the package.
- Prerequisites and interactions are appropriate to student ability and maturity.

VI. **Final decisions and purchase:**
- Compile results from previews.
- Select the winning package

Bibliography

A listing of the more comprehensive and useful readings.

1. Cohen, V.B., "Computer Courseware Development and Evaluation." *Educational Technology*, 23:1 (Jan. 1983), p. 9-14.
2. EPIE & Consumers Union, "No Preview, No Purchase." *The Computing Teacher*, 10:5 (Jan. 1983), pp. 35-36.
3. Forman, D., "Courseware Evaluation for Curriculum Integration." *AEDS Monitor*, 20:10, p. 10.
4. Jostad, K., "Search for Software." *AEDS Monitor*, 20:10, pp. 25-26.
5. Lathrop, A., Goodson, B., *Courseware in the Classroom*. Menlo Park: Addison-Wesley Publishing Co., 1983.
6. MicroSIFT, *Evaluator's Guide*. Eugene: ICCE, 1983.
7. Neumann, R. "A District Software Evaluation Plan that Works." *Electronic Learning*, Oct. 1982.
8. NCTM, *Guidelines for Evaluating Computerized Instructional Materials*. Reston, VA: NCTM, 1981.
9. Slesnick, T., "Hold It: You're Using Computers the Wrong Way." *The Executive Educator*, April 1983, pp. 29-30.
10. Steinberg, E., "Reviewing the Instructional Effectiveness of Computer Courseware." *Educational Technology*, 23:1 (Jan. 1983), pp. 17-19.
11. Olds, H. "Evaluating the Evaluation Schemes." *Evaluation of Educational Software: A Guide to Guides*. Chelmsford, MA: NEREX, Inc., 1983.
12. Watt, M. "Making a Case for Software Evaluation." *The Computing Teacher*, 9:9 (May 1982), pp. 20-22.

[Anne Batey is a graduate student in computer education at the University of Oregon. She has taught high school science in North Bend, Oregon for the past five years.]

© Copyright 1983 ICCE

Enrichment Courseware for Middle School Mathematics

by
Lois Edwards

Computer courseware is a natural resource in middle school mathematics classes. Since middle school mathematics students possess widely differing degrees of motivation, skill in basic arithmetic, experience in applications and problem solving, and maturity toward abstract reasoning, teachers should try to provide individualized instruction over a large range of topics. Fortunately, middle school mathematics classes, ranging from remedial arithmetic to first-year algebra, are often less tightly structured than the elementary arithmetic curriculum that precedes them or the traditional algebra and geometry sequence that follows. This gives teachers somewhat greater flexibility in selecting content, ordering topics of study, and varying the delivery of instruction. Courseware can effectively deliver traditional content and can effectively introduce students to math concepts not otherwise accessible.

Motivation is of primary concern at this grade level, both for students who have not yet mastered elementary arithmetic and for talented students who may be bored with arithmetic. Courseware can provide engaging learning experiences for both of these groups, as well as for average students.

Tutorial and drill courseware with traditional content has been well-publicized, so this article concentrates instead on representative programs that address problem solving, applications, number theory, probability, number systems, graphs and pre-algebra topics.

PROBLEM SOLVING

For best results, problem-solving courseware should be integrated into a problem-solving unit with other learning activities such as teacher presentations, manipulatives, class discussions, etc.

Problem-Solving Strategies, from the Minnesota Educational Computing Corporation (MECC), contains four programs to teach the strategies of trial and error (guess and check), exhaustive listing and simplifying the problem, along with tools such as making a chart and looking for patterns. The content consists of mind-teaser, puzzle problems: How many squares are there in a 5x5 grid? (not just 25!) How can one efficiently color a map? What relationship exists between the size of a pool table and the path of a ball on it?

Two tutorials teach the three strategies directly with color graphics and a high degree of student interaction. Then, in *"Thinking With Ink,"* students try to color a map at the lowest possible cost with paint colors that have different prices. Simplified warm-up maps and complex challenge maps are provided. Students will keep coming back to improve their previous scores by devising even better strategies (there is no "right" answer). The fourth program, *"Pooling Around,"* a simulation of a pool table with one ball, enables students experienced in problem solving to collect data and find a variety of patterns relating the pool table dimensions to the path of the ball. This is best used as a small group activity, so that the exchange of ideas and verbalization of hypotheses increase the instructional value.

The Factory, by Sunburst Communications, focuses on the strategy of working backwards. After experimenting with a variety of simulated machines which punch holes, paint stripes and rotate objects to discover their effects on a square of metal, students are challenged to design a series of machines that will duplicate a computer-selected finished product. In another option, a student can specify a finished product and challenge classmates to create the required machine. Excellent color graphics contribute to the intrinsically motivating, deductive-thinking exercise.

Another package from Sunburst Communications, *The Pond*, works with pattern recognition—patterns of lily pads in a pond with a pad-hopping frog who tries to stay dry. While the simplest levels of this program are easy enough for young children, the most advanced will challenge adults. This package provides practice in finding visual patterns (two pads up, three left, etc.) that should help students see numerical patterns in mathematical problems.

Rocky's Boots, from The Learning Company, combines problem solving and creative thinking with a content area not normally found in middle school—logic gates (and, or, not). Excellent color graphics and a high degree of student control create a motivating environment in which students devise machines using wires and logic gates, employing both analysis and synthesis while they enjoy the games. Four programs teach the skills needed to play the two games included in the package. In a mathematics class, this courseware could be tied into studies of Venn diagrams or formal logic. It is good preparation for computer programming.

Deductive thinking is the basis for two educational math games that have demonstrated their effectiveness over the years. In *Bagels* the program selects a random two- or three-digit number which the student tries to guess from clues about correct digits and placement. Interpreting the meaning of the clues and selecting the best guess is both intellectually challenging and fun. *Taxman* requires an understanding of factors of whole numbers and provides a challenge to students to test their knowledge of primes. The Taxman gets the factors of the numbers the student chooses, so students do a great deal of "what if" thinking to devise a strategy to beat the Taxman. *Bagels* and *Taxman* are found on MECC's *Elementary Volume I* and in numerous other collections.

In *The King's Rule*, from Sunburst, students form hypotheses to deduce what rule fits a given set of three numbers. The six levels of difficulty provide an appropriate challenge for any student. In creating several hypotheses for a given problem, students must distinguish between what must be, what could be and what cannot be. This highly addictive program keeps groups of students (and teachers) thinking and

© Copyright 1985 ICCE

debating their hypotheses until someone turns off the computer.

Magic squares and related arithmetic puzzles have been used for many years to stimulate pattern recognition. The computer program *"Magic Squares"* in the *Arith-Magic* package from Quality Educational Designs (QED), guides students through explorations of averages and sequences. Two additional problem-solving games involving subtraction and multiplication complete the package. Magic square puzzles and similar challenges can be generated and printed on paper using MECC's *Computer-Generated Mathematics Materials, Volume 1.*

APPLICATIONS

Applying arithmetic and problem-solving skills to real-life problems is aided by several different types of courseware. One such aid is simulations, like the economic simulations *Sell Lemonade* and *Sell Bicycles* (MECC's *Elementary Volume 3).* Simulations present realistically complex situations using random numbers and the fast and accurate calculations of the computer. Students can practice interpreting mathematical data, analyzing situations and deciding on a strategy to solve the problem. The computer alleviates the burden and frustration involved in complicated arithmetic calculations.

The simple database in MECC's *Polls and Politics* courseware enables students to collect data on a questionnaire of their own design, and then enter it into the computer. The program calculates and displays frequencies of responses and cross tabulations in both raw data and percentage figures. Students learn to read, interpret and analyze data through a realistic example of the value of percentages in a statistical setting.

Word problems, a perennial source of difficulty for middle school students, can be investigated and solved using an innovative program called *SemCalc* (semantic calculator) from Sunburst Communications. Students enter each item given in the problem, specifying *how much* (numerical value) and *what* (unit of measure). Then they decide what operation to perform. The program displays the units of the answer produced by the student's choice of operation. If these units don't match the desired ones, the student knows to try a different operation.

NUMBER THEORY, PROBABILITY, NUMBER SYSTEMS, GRAPHS

Factors and primes, important topics in working with fractions and later in algebra, are given a mathematically sound and highly interactive treatment in the *Factoring Whole Numbers* courseware package from Quality Educational Designs (QED). The novel format of a tutorial and a game or exploration activity for each topic (12 programs in all) provides a motivating, concrete approach to factoring concepts, including the Sieve of Eratosthenes, twin primes and Goldbach's conjecture. The exploration programs provide challenges in discovering patterns in number theory.

Two other QED packages, *Fractions* and *Decimals,* use this same tutorial in a game/exploration format and are designed to follow in sequence. They encourage the use of manipulatives along with graphic computer tutorials. *Decimals* includes games involving decimal/binary conversion.

Computers are a valuable tool in teaching probability because of their ability to simulate random events and to perform large numbers of calculations quickly. MECC's *Probability* courseware includes both simple coin simulations and a surprising game activity that motivates a deeper study of expected events in rolling dice.

While number systems of bases other than ten have long been a topic of study because they increase understanding of our base ten system, computers provide a real-life example where other bases (two and sixteen) are important. The *Number Systems* package from MECC includes complete lesson plans plus computer programs to provide practice in converting between bases in a game setting.

MECC's *Graphing* courseware combines the math topic of graphs with the computer's graphics capabilities, creating a bridge between math and programming concepts.

PRE-ALGEBRA TOPICS

Discovering Equations, from Sunburst Communications, uses balance scales to introduce the concept of solving simple linear equations, demonstrating the effectiveness of the computer as a manipulative in teaching abstract concepts.

Another Sunburst product, *Tobbs Learns Algebra,* has large enough print to be used effectively as a whole class demonstration activity. Students find numbers to fill in a chart according to a given relation, or they produce the relation to match the given numbers.

The popular game *Hurkle,* found in MECC's *Elementary Volume 1,* has been widely used to strengthen the concept of ordered pairs as points on the coordinate grid.

• • •

Middle school mathematics is rich in motivating, effective and instructionally sound computer courseware. Try incorporating courseware into your lesson plans and see how enthusiastically your students respond!

Note: The courseware described above is widely accepted and has been used by the author in inservice training workshops for mathematics teachers—however, mention of a courseware product in this article does *not* imply endorsement of it by the author or by any organization. END■

[Lois Edwards, Instructional Coordinator, Minnesota Educational Computing Corporation, 3490 Lexington Avenue North, St. Paul, MN 55112.]

© Copyright 1985 ICCE

Computers in Science Education

Richard C. Adams

For Heaven's Sake

Astronomy is one of those subjects that doesn't have a lot of class; its place in high schools is usually not too strong. Oh, I know there are wonderful schools out there with 15,000 students who teach Astronomy I, Astronomy II, College Level Astronomy and Astrophysics Seminar. But my high school has 485 students in it and is considered a medium-sized school in Oregon. I kind of think there are a lot of schools of that size or smaller out in the hinterland.

Astronomy is one of those subjects that may be taken up descriptively in junior high school science as a special topic in a minicourse, or perhaps in senior high school as a unit in earth science or general science. Occasionally, it sneaks into physics, but usually only if the teacher has had some background in it.

In my own case, I had taken both undergraduate and graduate college-level astronomy courses, and with my hobby of stargazing (much more fun than pure mathematical astronomy), I wanted to share this interest with my students. One of my students got so turned on to the stars ten years ago that he now works at the Very Large Telescope Array in Socorro, New Mexico, in radioastronomy. Things like that can keep a teacher going for several Mondays at least.

The point of all this is that, although astronomy can have intrinsic attraction, to present it properly requires some expertise in the sources. Giving an interested student the line, "Here's a book. . ." may quickly short-circuit budding curiosity. However, this is a column about computers in science, and I have a few solutions.

THE ASTRONOMY DISK
By Sheridan Simon
Prentice-Hall, Inc., Englewood Cliffs, NJ 07632
$29.95
Disk for Apple II series
Documentation

About two years ago I was doing some beta testing of computer programs for a local software company. Sheridan Simon had submitted two disks, one on physics and one on astronomy. The local software company delayed too long in replying to Dr. Simon, however, and Prentice-Hall got the first bid.

Dr. Simon is not only erudite and precise in his science, but he also has a sense of humor that comes through strongly on the disk. Some of his explanations, while sticking mainly to the point, occasionally go off on little flights of clever remarks and some wild allusions. The opening graphics are a start, but the rest of the humor is in the reading.

I was so impressed with his writing style that I got him to share some of his thoughts in an article in TCT ("Unprintable Physics," *The Computing Teacher*, Vol. 10 #7, March 1983, p. 14).

"Designing solar systems, planets and stars, and then letting the whole thing run for a few thousand years is just not in the budget of most science departments this side of heaven."

The Astronomy Disk is not the latest in graphics. Dr. Simon has chosen to use graphics only in those circumstances where they enhance the program and can help to get the point across.

Broken up into three groups, "Space Travel," "The Solar System" and "The Stars," there are 16 separate programs on the disk.

"Space Travel"

Many teenagers have watched all the films and know that spacecraft and "X-wing fighters" zoom around space with dives, curves and hard banks. With the aim of coming a bit closer to reality, "Space Travel" uses graphics to allow students to try orbits of satellites, design a multistage rocket, figure out the starting points of a voyage to Mars (where both origin and destination are moving in their own orbits), and even see how constant acceleration changes the nature of time on long flights. Programs like the Mars flight have no calculations, but you have to choose a particular moment to fire the rocket's engine. Students who tried this needed about 10 minutes to get a feel for orbit-to-orbit plotting. The mathematics can come later, after they have an urge to know more.

"The Solar System"

"The Solar System" involves visual representations of the satellites of Jupiter, planetary/satellite systems you design, the motions of comets and an orrery that shows the planetary positions, as seen from any angle, for any year between 1500 and 2500.

"The Stars"

"The Stars" again uses graphics to show things hard to present in a book. You can view a galaxy from any angle, see how a star changes on the interior, build up graphs of radiation output depending on temperature or spectral class—and even build a world. This last program was one of the most popular with my students, but I feel an important part is now missing. Given choices of stellar class, stellar type, size of planet, albedo, mass of planet and distance from the star, the program figures out what the year and temperature would be. Students played with adjusting orbits slightly or adding reflectivity (albedo) to the atmosphere and the like, trying to get a terrestrial environment, but in wild solar systems at impossible distances. At that point the original program used to then draw a scene from the planet, showing how big the star would look in the sky, the size of mountains expected with that gravity, and snow, green grass, fried rock or molten surfaces, depending on temperature. My students liked this part the best because they could then stand on their planet and look out the window. I miss the fact that the new program has no such graphics. Perhaps Dr. Simon can be enticed to put them back in and remove something else.

Sheridan Simon has used these programs in his college level classes at Guilford College, Greensboro, North Carolina, as assignments done without teacher assistance. To really get the most out of these programs, about one hour should be spent trying out the variations and fiddling with the parameters. Thus, with 16 programs, you can look forward to about a week's worth of work and a

© Copyright 1985 ICCE

good minicourse in astronomy.

This is one of those cases where the power of the computer can be used to simulate a lab that would be very difficult for most schools to set up. Designing solar systems, planets and stars, and then letting the whole thing run for a few thousand years is just not in the budget of most science departments this side of heaven.

TELLSTAR: YOUR COMPUTER GRAPHICS WINDOW TO THE UNIVERSE

Scharf Software Systems, Inc., Suite 1068, 2111-M 30th St., Boulder, CO 80301

$79.95 Apple; $99.95 IBM; $3.00 additional shipping and handling charge

Apple II series disk (also for IBM)

Tellstar is something else. It is not meant to be a tutorial in any sense of the word. Instead, it is a tool for stargazers. True astronomers never get to look at stars; most of the time they look at strips of film showing spectra of stars to see minute changes in distance, speed or rotation. In fifth through twelfth grade-type astronomy, you need to take the class to see stars at least once. Here in western Oregon, I've had to pick my times very carefully to see any sky at all during the winter.

What *Tellstar* does is give you your own private planetarium. It's not as good as seeing the big one with all the lights out and the kids going "Ahhh!" However, it *can* prepare students for their field trip or what to look for each night, even if the skies maintain a constant drip. Using this program, you can see what the night sky will show for any location, any date from A.D. 0 to 3000, any time of day (it will even show the sky as it would look during the daytime, complete with sun), and looking any of eight directions to the horizon or overhead.

Stars, planets, Messier objects and constellations are shown. Moving cross hairs to any object will result in a full identification, including name, phase (planets), magnitude, location in right ascension and declination, time of rising and setting on that particular date and headings. You can move forward or backward through the calendar or the clock to see planets "wander" (as the Greeks thought in their name, "planet"), or see the constellations change with the seasons. Opportunities are available to print out the data on any object or even the sky map.

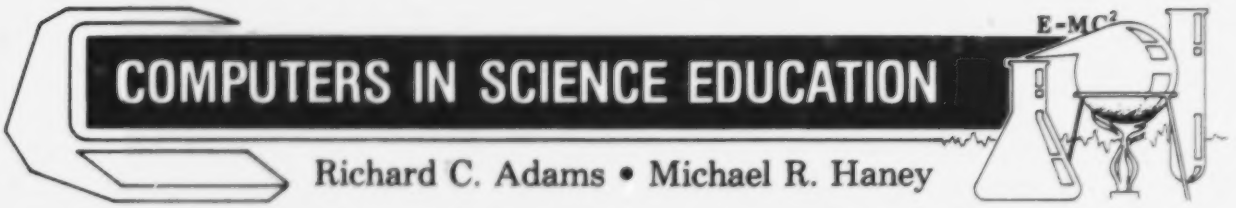
The program supports the graphics capabilities of the Apple Silentype printer (I personally don't know anyone who has one of those, of the 230 people I know who have Apples), but you can save the sky pictures on any initialized disk for later printing using the graphics dumps for *your* printer.

The 70-page manual contains numerous appendices with data on stars, Messier objects, planets and longitude and latitude readings for 63 North American cities and 39 cities in the rest of the world.

This disk probably won't teach kids as much astronomy as the one by Dr. Simon. It will, however, permit students to derive some things by tinkering with time and will allow the teacher to generate interest in field trips to the sky. ■

[Richard C. Adams, Pleasant Hill High School, 36386 Highway 58, Pleasant Hill, OR 97455.]

© Copyright 1985 ICCE



Doing Science

by
Richard C. Adams

During the year, I, as the reviewer for the Science column, receive software to review. A few months ago I stated I was interested in people using computers to **do science**, rather than telling about it with "automatic page turners." The following represent noteworthy examples of "doing science" software.

YOUNG PEOPLE'S MEDICAL SKOOL
Simulation Software Co.
6035 N. Maplewood Ave.
Chicago, IL 60659
Grades 7-12

Apple II + or IIe
$27.95 (shipping included)

This is one of those programs that you suspect will be terrible. Is there a glossy brochure? No, my copy arrived with original printouts from a dot-matrix printer. Then there's the name—"skool." Really, now. Does the disk have a four-color label? No, it almost looks typed. The disk isn't even copy protected, so you can make back-ups before you use it in a hazardous environment (the normal classroom). What can you possibly expect from something as poorly presented as this?

What you get is a rather nicely done simulation of medical diagnosis. I was pleasantly surprised by how well the graphics were done and even the inclusion of spoken-word sound effects. I've tried out this disk with three biology teachers, one of whom teaches Health Occupations, and ten students; we all had a lot of fun with it.

Here's how it works. You get to be the doctor and are presented with some symptoms. Some of these are in the form of notes from a patient's medical history, some are by graphical displays of the symptoms such as bleeding or rashes on the screen (in a tender location), and some are from words spoken through the Apple speaker.

You then select further lab tests you may wish to do from a menu. You can keep doing lab tests until you arrive at a diagnosis or the patient runs out of money. (The bank balance is prominently displayed.) If you do too many tests, the patient occasionally gets disgusted and leaves. S/he may also die before you figure everything out.

If your treatment involves surgery, you are given a choice of anesthetic and how much to administer. Vital signs are read out and most actions are displayed as rather good pictures on the screen with a lot of clever shadings (giving about 100 different colors on the normally 7-color Apple graphics). In one case, I suspected appendicitis and had the abdominal cavity laid open at which time I was given my choice of which organ to remove. After suturing, a progress report was given.

You can run into difficulty and your cure may not work. You then have to keep trying other things. Occasionally you'll lose one of your electronic patients. A biology teacher and I were trying to cure a male patient of a condition where his vital signs kept going down, no matter how we varied his diet and tried other treatments. We were so interested in the next move that the other teacher leaned forward to make a drastic change when her foot knocked the computer power cord out of the wall. All life-support systems were then shut down and we'll never know. . . .

It's fun to work on a medical "detective story"—and you also learn a lot. A list of medical terms is given in the somewhat skimpy documentation, but you are encouraged to use a library for more help.

There is a Part II for sale at about the same price ($27.95) and an easier version for younger children at $42.25—neither of which I have reviewed.

EXPERIMENTS IN HUMAN PHYSIOLOGY
(Catalog #502-00-SASA)
HRM Software, a division of Human Relations Media
175 Tompkins Avenue
Pleasantville, NY 10570
Grades 9-12

Apple II + or IIe
$249.00 with 30-day refund option

That price of $249 is kind of budget-straining when you first look at it. On the other hand, for that

© Copyright 1985 ICCE

price you get one disk, two temperature probes, an interface box, a timer plug, a light and sensor attachment, four extender cables, ribbon cable and a 75-page manual. (A back-up copy can be made.)

What does it do? The manual contains ten experiments dealing with human physiology. These are not *simulations* of human measurements; rather, you do the measurements on a human subject and then interpret the results. Reaction times of various parts of your body can be measured and temperature differences of different parts can be taken. Even your heart rate can be measured in a quite clever manner, with a light shining through the skin and a sensor checking to see how much light is transmitted or reflected, depending on how much blood is going by that point at any one time. The screen shows both immediate readings and long-period readings of the rate, so biofeedback techniques can be developed and slower responses also charted.

All of the probes attach by color-coded cables to an interface box with eight probe sockets. The box is plugged into the game paddle port inside the computer, but an extra game paddle port is on the box so you don't have to unplug everything if you want to use the port for something else in the meantime.

Using the equipment, you can even measure breathing rate by putting the temperature probe near the nose of a subject and noting the change in temperature on exhalation and inhalation. You can rig up a polygraph for "lie detection" should the need arise. Athletic departments can use the devices for conditioning measurements quite easily. The chemistry and physics departments can use the temperature and light probes rather nicely also, making the $249 price less costly if split among four or five departments who take turns using the equipment.

A demonstration disk is available which basically shows menu layouts and the like, with the response-time experiment actually included, since it does not require any of the hardware. Response time is measured after a stimulus of a sound or a flash on the screen.

I have not seen the whole package, but the demonstration is impressive. I conferred with a colleague who had purchased the package, and he had had good results with it and found very few problems. It *is* a lot of money (at least for my budget), but as I said, perhaps some sharing can be done. The manual should get you started with a number of intrinsically interesting experiments, but the package seems to encourage you to develop more uses, once you have precision equipment available. You may even discover that one shared package is not adequate, once you find out all you can do with it.

Producer's Reply:

We appreciate the opportunity to respond to the review.

The demonstration disk is available free on a 30-day preview.

Also available from HRM Software:

- *Experiments in Science*, a series of 12 experiments in biology, physiology, chemistry, physics and earth and planetary science. Hardware is also included in this package.
- *Health Awareness Games*, a set of 5 programs that draws on extensive tables of statistics on people's health.

MICROCOMPUTER PHYSICS LAB—HEAT
Cross Educational Software
P.O. Box 1536
Ruston, Louisiana 71270
Grades 10-12

Apple II + or IIe
$60.00

On a slightly lower scale, in both price and execution, we have the programs from Cross Educational Software. Dr. Mark Cross has written articles about educational philosophy in *Call A.P.P.L.E.* and elsewhere and has a firm foundation in physics as well.

Rather than an all-inclusive program dealing with many measurements, Cross has taken the method of providing modules. The one I reviewed was dealing with heat. There is another one dealing with light which measures the time interval between flashes of light or something blocking the probe, such as a cart moving down an inclined plane or a pendulum. For $60.00, you get a diskette, a three-ring binder for documentation, and the necessary hardware for four temperature probes. To use them, however, you have to put them together, which is not really that hard. It involves soldering four wires onto a 16-pin plug and the other ends to two thermistors (which look like little blobs of metal with two wires coming out). Everything is provided except the solder and a soldering pencil.

Rather straightforward and simple, the programs on the disk allow you to calibrate the thermistors (each one will be slightly different when you get them hooked up), get a continuous readout of temperatures on the screen, or, most importantly, see the temperatures displayed both in digits and in a nicely done graph on the screen. The range of the temperature graph can be set as narrow or wide as you want, and the time interval between temperature readings is adjustable between one and 1000 seconds. That means a graph can display 270 seconds (4.5 minutes) or as long as 270,000 seconds (75 hours). You could, for example, set up a heated object in a box with some specific insulating material and come back the next day (or three days later) and see how the temperature had dropped. A second thermistor could be reading the room temperature's variation at the same time and a third just be a control in a box of insulation. (A fourth thermistor is provided if you want to hook it up.)

© Copyright 1983 ICCE

Your graph would show all three lines in a quite readable form on the screen. If you wish to have a hard copy of the numbers, just turn the printer on before you leave and the numbers will be listed there instead.

The program could be easily modified to write to a file on the disk, although timing might be changed slightly.

You can do that sort of thing because Cross Software has no "protections" on the disk. You can copy it or modify it as you see fit. This program can easily be used by a student without much preparation, but full usage of this package assumes there is a real teacher out there who is willing to experiment with new applications. The manual gives outlines of five experiments for use and also includes listings of all programs used on the disk, including the assembly language timing routine, for you to learn from and modify.

I had a problem with the program when I first received it. I wrote a letter to Dr. Cross and had a new, corrected version back in the mail within five days. A letter accompanied it that showed I was dealing with a real, live person who cared about how his software worked for teachers.

Again, although this software does only basically one thing, it does it well and at a reasonable price.

Also available from Cross Educational Software: 12-volume *Physics* sets ($260.00 total or $15-$35 each); and *Physics Lab-Light* and *Physics Lab-Sound*, both at $60.00.

PRECISION TIMER $39.95
GRAPHICAL ANALYSIS $24.95
RAY TRACER $24.95
Vernier Software
2920 S.W. 89th Street
Portland, Oregon 97225
Grades 10-12

Apple II + or IIe

David L. Vernier has taken the "module" approach, similar to that of Dr. Cross. In this case, however, the equipment does not come with the software.

The disk called *Precision Timer* is basically a timing device to tell when one of the paddle buttons has been pushed or released. In the operation of this disk, the activation is by light, rather than by pushing a button, although it could be modified to do that.

The disk is designed to operate with "photogates" in a module from PASCO (part #ME-9204). Each of those gates costs $75.00 and you will use two of them. An interface box to the game port costs $100.00 (part #AI-6575). On the other hand, you can build the interface box with *three* photogates using parts bought at Radio Shack for $24.12. Mr. Vernier gives instructions on what parts to use and how to put them together. I know how to solder but am weak on semiconductor devices. Even so, I was able to put together the whole thing in about three hours. The worst part was going around to five Radio Shack stores to get the 18 parts, since each store seems to only have one of each item. Phototransistor FPT-100 usually consists of a card on a rack with "Sorry, out of stock" right behind it. Other than that, putting the hardware together is fairly straightforward.

Once you have everything together, the software does the rest. There are 12 different modes of measurement, depending on how many light sensors you wish to use. Basically, each works by cutting off light to a very small sensor. This begins the timing, and then turning off another or the same one again stops the timing. You can drop a ball past two sensors and get the speed, past three and get the acceleration, let a pendulum go past one sensor twice and get the period, perform collisions with carts or airtracks and get all the data you need. You can also use the keyboard, without any other equipment, to make this program a quite accurate stopwatch (+ or − 0.0002 seconds).

One nice feature of the disk is that all numerical displays can be shown on the screen in numbers four times as big as the normal digits on your monitor. Thus the data collection can be seen from the back of the room for demonstrations or larger group labs.

One caution about this equipment: the third light sensor uses paddle button #2, which is also the pin used in most "shift key modifications" for the Apple. That means that it will appear that the shift key is being held down. One way to do the shift key mod is from the #1 pin of the chip just in front (toward the keyboard) of the game paddle port, to the keyboard #2 pin using spring clips (Radio Shack Cat. #270-372). Installation takes about 15 seconds, but more importantly for this experiment, *disconnecting* it can be done in about 5 seconds.

Up to 100 data points can be collected, and then average and variance will be printed out. For many of the experiments, further mathematical treatment is possible, specific for the usual use. For example, the pendulum experiments will give the data of period, frequency, etc. that you would normally calculate.

The 42-page manual contains a user's guide, a teacher's guide, modifications to the equipment and programs that can be made, and a complete listing of all programs with a table of variables. The disk is completely unprotected so back-up copies and modifications are easily made.

Graphical Analysis can interface with the data from the above experiments, collecting stored data from the disk directly. Or, you can enter pairs of data for up to 32 points.

The disk is a very quick way to visualize data. Putting in the points is one option. Using the Plot option, you can label the axes, put in the units to be displayed, plot just points, points with connection,

© Copyright 1985 ICCE

or "point protectors" of boxes around each point. A linear regression analysis can plot the closest fit straight line also. Other menu options give you the choice of exchanging the axes, changing any pair of points, and a numerical analysis section giving slope and intercepts.

The program doesn't do anything really fancy, but for $24.95 you get a very quick way to display lab data in a graphical form. The scales on the side can be done in "auto-scale" mode for quickness, or you can determine the value of each unit for clarity. Although this is not as instructive as learning how to make a neat graph with ruler and pencil, perhaps your interest is in teaching how to *set up* graphical displays and what they *mean*, rather than the mechanics of pencil and paper neatness.

Ray Tracer is an electronic blackboard. There is nothing on this disk that a good physics teacher couldn't do with a blackboard, chalk, straight-edge, protractor, compass, fantastic skill in drawing and three hours for each class period.

The key differences are in accuracy and time. I can draw a good picture of what a light ray will do at the interface between air and glass. But I can't do it accurately for both red and blue light, for any pair of materials, and for 15 different angles in under two minutes. I can *talk* about the "critical angle" where light bounces off of water, but it's more instructive to "sneak up" on it by changing the angle of incidence slowly until the reflection takes place on the screen.

Besides refractive indices between two media, other options from the menu of 15 situations deal with thin and thick lenses, compound lenses, plane and spherical mirrors. In each case, one white ray will be drawn, or you have the option of doing two rays (red and blue) to show the difference wave length makes. The programs can be run by students as a "game" since a source is picked arbitrarily and then a target is placed somewhere. The object is to hit the target, no matter what media you have to go through or reflect from or refract through. This sort of "gaming" can teach a lot of intuitive feelings for what optical devices do without a lot of difficult lab setups.

The 29-page manual has a user's guide, a teacher's guide, what optical laws are being demonstrated and a list of experiments and references. A complete listing of the programs and variables used are also included on this unprotected, LISTable disk. You are given hints on modification and are encouraged to do so, to make the programs fit your exact requirements.

If the drawings aren't fast enough for you, there is also a *compiled* version on the disk that runs 2-3 times faster, although modification or even listing is then impossible.

Again, at $24.95, this disk doesn't allow you to do anything you couldn't do before, except improve accuracy, save time, and make optics more interesting. Perhaps such things are attractive to you.

Also available from Vernier, but not reviewed: *Orbit, Projectiles, Kinematics, Charged Particles, Vector Addition*, and *Wave Addition*, all at $24.95. (These programs have also undergone recent revision.)

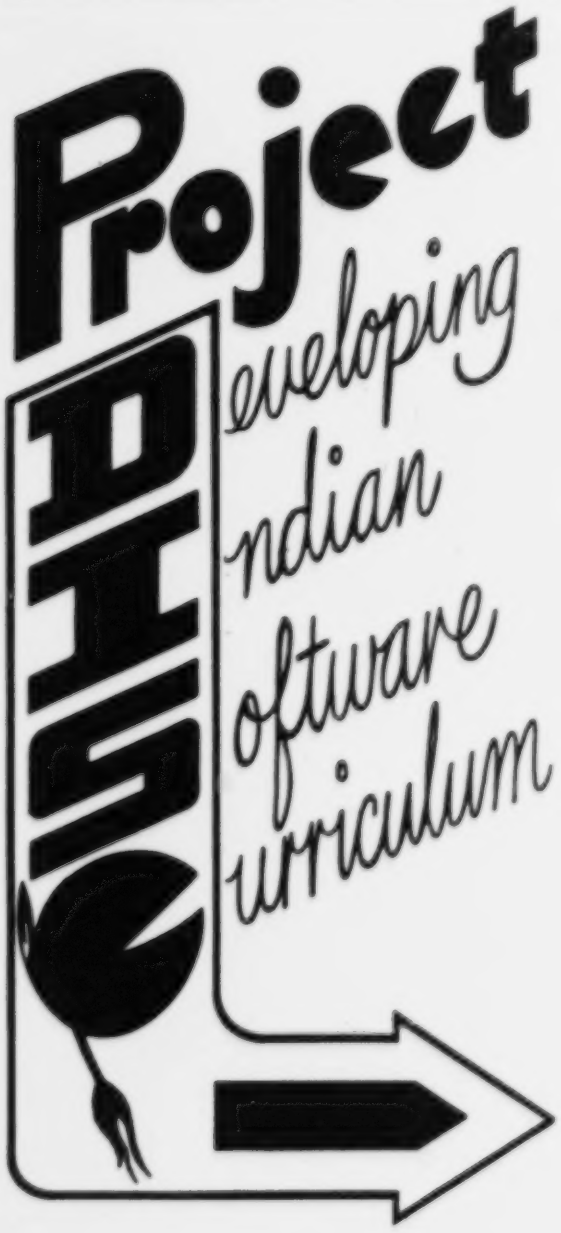
Author's Reply:

Graphical Analysis has undergone major revisions with several new features added for the same price.

- 200 pairs of data may now be entered, instead of 32.
- Data is rounded before it is displayed—the number of significant figures is controlled by the operator.
- The appearance of the final graph has been improved.
- Semi-log and log-log graphs may be displayed.
- A hard copy of the graph can be made on any standard 80-column printer.

END■

© Copyright 1983 ICCE



"I did a good job on nouns and verbs today. Could I try adjectives tomorrow?"

"We need another new dictionary, Mrs. Jacobi. This one got worn out!"

"Billy is hogging the computer again!"

"Boy is this a neat piece of software! Can I come back and finish it after school?"

"Mrs. Jacobi, I just LOVE computers!"

by
Carolyn Jacobi

In July of 1982 a three-year Title VII demonstration grant was awarded to the Rapid City School District. It was the largest grant awarded that year, and its reason for existence was to determine whether or not computer assisted instruction could help Native American children. Two and one-half years later, we think we have some answers.

Statistically Speaking

by
Carl Edeburn

My experience with this project has been truly rewarding. As a teacher-educator I have been involved in and intrigued with the potential of CAI since the early 1970s. However, experience in research and evaluation has caused me to look with some skepticism upon the blind rush to infuse computers into all aspects of education without a collaborative effort to measure the impact. Working with this project has allowed me to "scratch the surface" in establishing a better understanding of the potential computers have for improving the educational environment.

The ultimate goal of this three-year project was to improve the reading and language skills of an identified group of Limited English Proficient Native American students.

Lakota (Sioux Indian) children constitute 13 percent of the district school population. In the first year of the project, efforts were focused in two elementary schools which contained the highest numbers of Lakota children. These included a school with a 55 percent Lakota population and a second site which had a 38 percent total. In Year II an additional site (16 percent Lakota) was selected for participation.

Since the use of such a technologically innovative approach was new in 1981, there was very little research in this area. Hence, the project was plowing new ground in the areas of curriculum development and pedagogy and had implications for baseline research as well as the fulfillment of the immediate educational goals. The purposes were:

1. To develop and incorporate into purchased educational programs instructionally appropriate materials;
2. To develop pedagogically appropriate programmed materials for use with microcomputers; and
3. To train instructional staff in the methods of CAI technology.

The students were chosen using a combination of methods including standardized test scores, assessment scale scores,

© Copyright 1985 ICCE

and a review of demographic information which included family income and teacher/principal input. A total of 108 students were involved in the project.

A dedicated, capable Computer Specialist is a must in a project of this type. Without Carolyn Jacobi's expertise as a teacher and her interest in both the practical and research elements of this project, my participation would have been less meaningful and the project would certainly not have been as successful as the findings reveal.

Although the project was monitored via several data sources in the evaluation process, my discussion is focused on pre- and post-test results in the Comprehensive Test of Basic Skills (CTBS) and the CAI activity file.

The CTBS subscale scores for Word Attack, Vocabulary, Comprehension and Composite Reading, as well as the score for Spelling, were retrieved from the fall and spring testing process to determine whether or not growth had occurred.

To address these dimensions, the Norm-referenced evaluation model (N-RM) was employed. In the N-RM, performance (or gains) of the treatment group is compared to the performance of the norming sample, i.e., the group from whose performance the test forms were derived. Therefore, the norming sample (national norms) are used as a comparison group against which to compare the growth or gains in the program group(s).

Fall and spring tests were administered within two weeks of when the norms were obtained. The norm-referenced comparison was implemented on the individual and composite reading scores, as well as on the spelling scores of the CTBS.

In the second year of the project, 11 (85 percent) of the 13 subscale by-grade comparisons evidenced significant growth, and all of the total group comparisons revealed significant growth with probability levels of < .001. The total group comparison is summarized in the following table:

Summary of Means, Mean Growth, t Values and One-Tailed Probability Levels by Subscale by All Grades

N = 108

Subscale	Fall NCE Mean	Spring NCE Mean	Mean Growth	t	P
Word Attack*	33.95	51.59	15.64	4.64	.000
Vocabulary	33.87	43.56	9.69	7.48	.000
Comprehension	37.09	44.31	7.21	5.36	.000
Composite	36.51	45.33	8.82	8.05	.000
Spelling	39.18	46.33	7.16	5.44	.000

*Grade 3 only (N = 22)

Background

After many years of "Indian programs" in the local school system, the Native American student was making no significant progress when it came time to interpret test scores each spring. This bothered Dolores Riley, then Title VII Director. Would another approach prove to be more successful? And, at the same time, would another approach provide these Native American children with in-roads into computer use? (Computer access at that time was limited primarily to students in schools where PTA funds or parents' clout could underwrite the school budget.)

Rapid City is a town of 65,000 people, located at the foot of the Black Hills. Mount Rushmore is 30 miles south and Bear Butte, the holy and sacred mountain of the Cheyenne and Sioux nations, is located 30 miles to the north. There are nine Indian reservations in the state. The reservation is the psychological and social/cultural center of the lives of many of our students. Although families often move to Rapid City to find work or attend school, weekends and vacations are spent on the reservation. And many students actually move back and forth from reservation to city several times in the course of the year.

Carolyn Jacobi, Computer Specialist; James Livermont, Title VII Director; and Lowell Amiotte, Director of Indian Education, greet newest staff member, TOPO.

In 1982 our district offered programming classes at the high school level, a Computer Literacy Committee was investigating the role of computers in the district, Chapter I personnel were using drill and practice tapes, and some enterprising teachers were bringing their own computers to their classrooms. In 1982 the idea of using computers in a lab as an integral part of some elementary students' instruction was an innovative idea indeed.

A needs assessment done in the target schools and in the community determined that area parents and teachers were most concerned with reading and language arts scores. Since aides were already working with K-2 teachers in the impacted schools, the grant was written for students in grades three through six. During the summer of 1982, introductory computer activities were planned, a computer specialist was hired, and workshops were held for teachers in the two most heavily impacted schools, where the first two labs would be placed. Computer aides were given two full weeks of training, and a "kick-off" day was held in early September for the staffs of both schools. In nearly every case, these workshops were the first computer experiences for either teachers or aides, and they were extremely valuable.

The Program

The labs operate similarly to each other, but there are some important differences. The manner in which children are tracked, their time on computers, and the types of software they use is very exact. Another constant is the make-up of each group of children brought into the lab. All groups have 60 percent identified students and 40

© Copyright 1985 ICCE

BEST COPY AVAILABLE

BEST COPY AVAILABLE

percent non-identified. This has had a positive influence on the success of the labs, because it has kept project students from being identified, either with a special minority program or with a remedial program. In fact, it has given us the ability to give time on the computer to many children, since most classroom teachers used the 40 percent of their allotted groups to rotate their other students through the lab. This allowed us to be instrumental in the computer literacy process of students other than the ones we served, and it gave our identified students the prestige of teaching "new" students in their class.

An area of difference in operation has to do with the way the children are scheduled. Working with the building staffs, it was decided to pull students out of class for this program, even though pull-out programs normally have a low degree of acceptance in this district. By tailoring students' time out of class to the teacher's schedule, the degree of negative response was cut considerably. Some staff schedule children on a rotating basis, some on a fixed time allotment—and one school has a mix of the two. Picking up students from their classrooms and delivering them on time makes the program more acceptable, also.

Students are in the program for a half hour each day. One day a week they are in a tracking/testing system, with aides keeping a careful watch for student needs and further work to plug in software practice. Three days a week students are given other pieces of tracked software and, if attendance and work are satisfactory, they have their choice of software on the fifth day.

The size of the t value is an indicator of the difference between the fall distribution of NCE scores and the spring distribution of NCE scores.

The P score is an indicator of how many times this difference (a t value of this magnitude) would be expected by chance alone in 1,000 replications of the study. As can be noted in the table, all of the differences (mean growth) were of sufficient magnitude to generate very small probability levels. Common sense would demand that in the case of the greater difference (t) and the smaller probability (P), the more convinced we would be of any real or actual changes.

The evaluation process incorporated a dimension which attempted to pinpoint those elements of CAI participation (treatment) which had the greatest effect upon growth in the three major reading scores (Vocabulary, Comprehension and Composite). The approach employed was stepwise regression, a statistical approach which is based upon the principle of correlation or co-variability between and among variables. These variables were the time each child spent interacting with the 10 different software packages employed in the project (predictors) and individual NCE gain (criterion) on the three subscale scores.

An interesting pattern emerged from these three analyses. Although three or four statistically significant software packages emerged from each of the regressions, one package emerged as the single best predictor in all three instances. This software package, *Microzine*, had characteristics which included programming, spatial relations, graphics (Logo-like) activities, and organizational sequencing, all of which are problem-solving "proactive" learning activities as opposed to the typical "reactive" CAI drill and practice routines.

As a part of the evaluation process, by-sex statistical comparisons were implemented. These analyses revealed no differences in the NCE gain scores of the students when compared by sex.

Many educators have indicated that there is substantial inequity among schools. According to Stone (*Teaching and Computers*, January 1985) and *The Computing Teacher*'s Equity issue, April 1984, schools in high income areas are characterized by the following:

1. More money.
2. Parental support of the technology (hardware, software).
3. Community support for teacher training and inservice efforts.
4. Students learn to direct the com-

© Copyright 1985 ICCE

puter through programming, data base management and word processing.

On the other hand, schools in low income areas are characterized by:

1. Less money for computers.
2. A constituency of parents who are either not educated or able to express their concerns about current educational issues.
3. Student use of computers (if they use them) largely restricted to drill and practice.

Stone indicated that the CAI approach in low-income schools is a result of the amount of teacher inservice. The implementation of drill and practice and other reactive software activities requires much less teacher inservice than do processes such as programming and problem solving.

An interesting parallel to this argument is the reactive approach to teaching writing in the language arts, where many schools rely upon worksheets or workbooks in the areas of structure, grammar, usage and punctuation in the hope that students will somehow "put it all together" when they attempt to write something. Other schools spend the time actually teaching processes of communication, including organization, exposition, and actual problem solving. *Instruction* in writing is much more demanding than processing students through a grammar workbook, and hence is more expensive.

This study tends to corroborate the value of software problem solving and inductive CAI activity in that they contribute positively to growth in both vocabulary and comprehension, **whereas some of the packages, characterized by drill-and-practice routines, were actually negatively correlated with the NCE gains.**

This finding may well make us question the efficacy of typical drill-and-practice software, and focus instead on incorporating problem-solving activities in the teaching of reading skills. In the third project year we are attempting to focus on problem-solving activities.

END ■

[Dr. Carl Edeburn, Professor, Division of Education, Box 2220, South Dakota State University, Brookings, SD 57007.]

Finding language arts CAI software was no easy task in 1982. We used what was available. We found a tracking system, *PAL* (Universal Systems of Education), which contained a detailed, comprehensive breakdown of objectives that satisfied the most structured language arts teacher. This tracking system also provided a printout which was given to each child at the end of each week to take to his/her teacher and family. The need for this exacting communication is no longer necessary, but initially it helped to establish program credibility. This same system is still in use—nearly the only one that is—but the reports now include only a weekly printout for our files. Teachers and parents are given one on request, and a copy is submitted routinely every nine weeks for inclusion in the language arts report card.

In addition to the careful documentation of the curriculum, countless hours were spent talking with individual staff members, parent groups and interested community members. A software review committee was established, made up of parents and teachers. This is no longer needed, either. Our curriculum is trusted. Many teachers in the four target schools request certain pieces of software to be purchased or used with students when a specific problem is discerned. In fact, a bilingual program has been requested in more schools than we can cover—a first for our district.

The Results

Test scores after the first six months of operation were "statistically significant" in one school and "educationally significant" in the second. In 1983 we added a third laboratory, and this past year we attained "statistically significant" growth overall. It is also significant that we have observed no sex-related differences in adaptation to and/or liking of the computer. The vast majority of both boys and girls like the computer and benefit from its use.

The happiest part of the entire experience is that it tends to confirm what good teachers suspected all along. We have always said (haven't we?) that children need to practice reading more to become proficient at it. Records of time spent on software packages measured against growth revealed that ***the software packages that appear to make the significant differences are those that deal with imaginative reading and decision making . . . LOTS of reading and decision making.*** I was delighted!

The software below (in the order listed) had the greatest impact on our students. The absolute best was *Microzine* from Scholastic. It gave students lots of reading practice, and they liked it a lot! After that was:

Gertrude's Puzzles, *Rocky's Boots* — The Learning Company

PIK-PEK-PUT, *Riddle Me This* — Data Command

Weird and Wacky World I and II, *Dragon Game Series* — Educational Activities

We used two other packages which I can't recommend, both because they showed little correlation with students' growth and because they were terribly expensive.

© Copyright 1985 ICCE

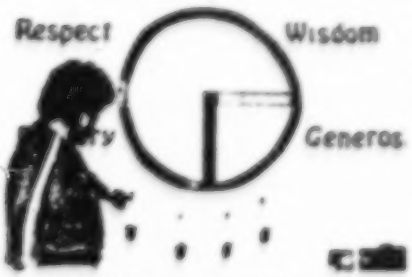
BEST COPY AVAILABLE

We also used a variety of other non-tracked software:

Code Quest (Sunburst)
Jigsaw (Island Software)
Arcademics (DLM)
Mapiown Hotel (Apple)
In Search Of . . . (Spinnaker)
Some MECC disks

The labs exchange software every nine weeks to keep material fresh.

The thesauruses and dictionaries really get a work-out!



Elaine Delores places display of the turtle's playground, Indian-style.

In the beginning, the lack of good educational software was the main problem. We moved out of the drill-and-practice mode as soon as possible and into tutorial packages. The software market is constantly monitored and more creative packages used whenever they fit the outline of the grant. Now the constrictions of the grant are the biggest source of frustration. Of late, the need to do more than CAI is evident, but we have been unable to move into the programming aspects of computing, and have used word processing only as a special, supplemental part of the program, even though nearly all of the aides have been trained in the teaching of writing, word processing and Logo. Many of the students now program in low-resolution graphics on the Apple. The combining of graphics and word processing with original stories of children just beg to be done and we are all excited with the possibilities, but the grant reads that each child will receive 30 minutes of computer time each day in language arts instruction, and we have adhered to that very strictly, documenting daily records of each student.

One "teacher pleaser" that I am happy to pass along is this: When you budget, be sure to include money for an ample supply of dictionaries and thesauruses. The amount of use they receive surpasses our wildest anticipation. The labs are given two copies of each at the beginning of the school year and these often must be replaced well before the end of the school year. This use of reference materials also speaks highly of the aides and their ability to know when to help and when to encourage student independence. The quality of personnel continues to be the main reason for success of this project, and program personnel are given many opportunities for recognition.

One area that required a lot of time during the first year of operation was the matching of paper and pencil tasks to the objectives being taught on the screen. The assumption was made that this was a necessary follow-up to the "real world" classroom situation. After the first year's analysis was made, the most significant factor to growth was the actual time spent on the computer. The paper and pencil activity portion was dropped the second year, and growth was even more significant the second year. We now feel safe in saying that our time and energy can be more beneficially used on the computer, rather than on doing paper and pencil match-ups.

The Lakota (Sioux) culture is in no danger of "technology takeover" in our laboratory schools. Each of the computer aides is Native American, and a concerted effort is made to keep the Indian culture very visible in the teaching materials and in the lab decor. Our logo, Iya, is taken from an old Sioux legend, and we are now anticipating making our TOPO robot speak Lakota.

Since the jury remains out on the keyboarding issue, we address it in a practical manner, i.e., we ask only that students use both hands on the keyboard—we make no attempt to teach formal keyboarding. It has been interesting to watch some near-classic styles emerge!

We are nearing the end of our three-year demonstration grant period, which is now in its dissemination phase. Unlike many projects, which simply *end*, the recommendations from our administration to the local school board will be that the essentials of the labs be continued in pared-down form.

The experience of this project almost defies description. My wish would be that every classroom teacher would have the opportunity to be involved with an honest attempt at educational research. It hones ideas, feelings and readings into sharply defined objectives and educationally defendable goals in a way I had not thought possible. And, if the efforts are labeled very successful, the taste of the icing on that cake is unbelievably sweet!

"Mrs., Jacobi, I am calling from Stony Plain, Alberta. We just purchased our computers, and your name was given to use as someone who might be able to help us." "I'm looking for a good disk to buy for Tad. He has some problems with reading comprehension." "We are looking for a speaker. Are you TOPO the Robot's mother?"

It's nice to be able to help sometimes! **END■**

Norman Blue Arm, artist

[Carolyn Jacobi, Computer Specialist, Bilingual/Bicultural Education, Rapid City Schools, 21 St. Joe St., Rapid City, SD 57701.]

© Copyright 1985 ICCE

Bank Street Writer— Does the Tool Define the Process or Vice Versa?

by
RANDALL BOONE

Word processing has become a new educational buzz word in the last couple of years, especially with educators who are wrestling with the issue of computer literacy in their schools. Student use of word processing software to compose, revise and print their writing is becoming standard in school curricula which offer computers. For teachers faced with the task of implementing computers into their classroom, it appears a safe, sure, first step into computer literacy for their students. This is perhaps true. Yet it seems that in this headlong rush to integrate computers into the curriculum, too little thought is being given to possible constraints which might lie hidden underneath the obvious advantages the computer provides.

Ask teachers about word processing and one program in particular, the *Bank Street Writer*, is likely to be a main topic of discussion. This word processing program, with its easy-to-learn, menu-driven format and secure handling of files, has virtually taken the educational market by storm. Teachers feel comfortable with this product; the name *Bank Street* itself implies educational excellence. The program is marketed to the educational community as being a facilitator of the composition process, and the idea seems sound enough. First, students write their thoughts without the hindrance of correcting errors or focusing on mechanics. When they have finished, then it is time to review the piece to revise both meaning and mechanics. This is the obvious advantage of word processing: changes can be made easily without the drudgery of rewriting the entire piece. Looking a bit deeper into the writing process, however, it becomes evident that *Bank Street Writer* may possibly impose serious constraints to a child's natural writing pattern.

Most writers will agree and indeed research confirms that writing is not a singular act, but rather an ongoing process consisting of several progressive stages. Rarely will any two authors' writing processes follow identical paths, yet three basic stages seem to occur in almost everyone's writing. The stages are most often categorized as 1) Prewriting, 2) Composing and 3) Revising. Research into the cognitive processes involved in writing suggests, however, that the writing process is not simply a linear progression through these stages. Rather, a model is defined which allows for interruptions during prewriting, composing or revising. Each of these stages is a recursive implementation of the same three stages. Revision and prewriting, for instance, may well interrupt composing, the middle stage, quite often. The reason for this is simple: writing is generative. That is, as one writes, new ideas are generated by the act of writing. This may cause a writer to want to make changes halfway through a story, a paragraph, or even in mid-sentence. It is at this point in the composition process that *Bank Street Writer* fails to accommodate the writer.

It would appear that the developers of *Bank Street Writer* chose a model of writing which requires the writer to compose an entire piece in one "stream-of-thought" burst of writing, not allowing for the recursive/generative nature of the writing process. Simple revisions during the composing stage are difficult to accomplish using *Bank Street Writer* with its separate modes for cursor movement and text insertion. The important act of revision is effectively discouraged by the complexity of the routine involved in accomplishing it. Word processing software which allows the writer to both write and edit in one mode is much easier to use and therefore more likely to be used for revision.

The following is a step-by-step description of this slow, seven-step process for backtracking, making a change, and returning to writing using *Bank Street Writer*:

1. Press ESC for cursor mode.
2. Move cursor to desired position.
3. Press ESC for write mode.
4. Make change or insertion.
5. Press ESC for cursor mode.
6. Return to previous position.
7. Press ESC to return to write mode and continue writing.

The alternatives are not any more appealing. One may simply backspace thereby erasing everything up to the point where the change is to be made, or hope to remember what change to make when going through the piece during the revision stage.

What is happening here is subtle yet very important. The way students naturally think during a writing episode is possibly being reshaped by the constraints of the word processor. This may be especially prevalent with beginning writers in the early grades. If this is true, educators must look very carefully at the implications this might have for students as writers and as thinkers.

There is little doubt that the ease of revision which word processing allows can be of great benefit for students as they begin to explore language through their own writing. As educators, it is our responsibility to provide them with the most appropriate tool for this task. It is necessary to examine word processing software with an eye for how much freedom the writer will have in using it rather than how easily it may be learned.

[Randall Boone is a doctoral student in computer education at the University of Oregon. He has taught elementary school for nine years, including five years overseas in Indonesia and Saudi Arabia.]

© Copyright 1985 ICCE